Lecture 4 : 행렬의 응용

Fastcampus Math Camp

신승우

Saturday 16th June, 2018

Recap on Previous Lectures

- 벡터/행렬의 연산
- 기저와 좌표계
  - 선형독립
  - Span
- 역행렬 구하기 / 연립방정식 풀기

행렬의 Column Space/Row Space

행렬의 column 벡터들의 span을 행렬의 column space, row 벡터들의 span을 row space라 한다. 행렬 A에 대해서 다음이 성립한다.

- elementary operation은 행렬의 row space를 바꾸지 않는다.
- A가 row echelon form이면, A의 row 벡터 중 영벡터가 아닌 벡터끼리는 서로 선형독립이다.
- A가 m by n 행렬이면, row space의 차원은 min(m,n) 이하이다.

Column space에 대해서도 비슷한 정리가 성립하며, 3) row space의 차원과 column space의 차원은 언제나 같다.

행렬의 Rank/Nullity

행렬 A에 대해서,

- 행렬의 row space의 차원을 그 행렬의 rank라고 한다.
- 벡터공간 $\{\vec{x} | A\vec{x} = \vec{0}\}$의 차원을 그 행렬의 nullity 라고 한다.

이 때, 다음이 성립한다.

- rank(A) $+$ nullity(A) $=$ (A의 row의 갯수)
- rank(A) $=$ (A의 row의 갯수)일 때, A가 invertible하다.
- nullity(A) $=$ 0일 때, A가 invertible하다.

선형변환

벡터공간 $V = \mathbb{R}^n$에서 벡터공간 $W = \mathbb{R}^m$으로의 선형변환은 V의 원소를 W의 원소로 대응시키는 함수 중 다음을 만족하는 함수를 말한다.

- $f(\vec{u} + \vec{v}) = f(\vec{u}) + f(\vec{v})$
- $f(c\vec{u}) = cf(\vec{u})$

선형변환과 행렬

선형변환은 행렬로 볼 수 있다. 더 정확하게는, 1) $V = \mathbb{R}^n$에서 $W = \mathbb{R}^m$으로의 선형변환은 m by n 행렬로 볼 수 있다.

Construction from Linear Transformation to Matrix I

V의 기저 $\{\vec{v}_i\}$를 생각하자. 이 때, V의 임의의 원소인 $\vec{v}$는 다음과 같이 쓸 수 있다.

$$\vec{v} = c_i \vec{v}_i \tag{1}$$

이 때, 선형변환 $f : V \to W$에 대해서 $f(\vec{v})$는

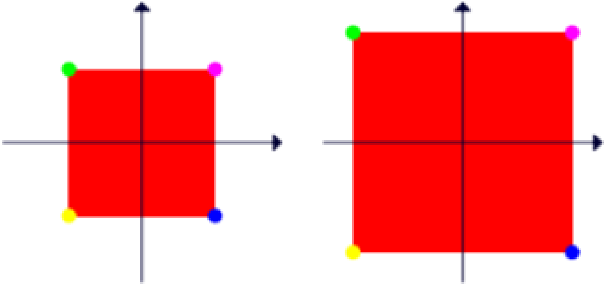
$$f(\vec{v}) = f(c_i \vec{v}_i) = c_i f(\vec{v}_i) \tag{2}$$

로 생각할 수 있다. 이제, W의 기저 $\{\vec{w}_i\}$를 생각하자. $f(\vec{v}_i)$는 W의 원소이므로 다음과 같이 쓸 수 있다.

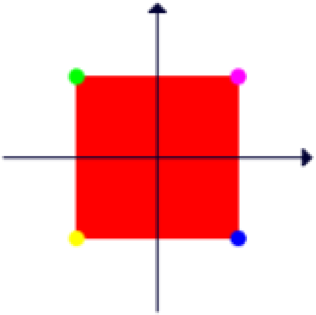
$$f(\vec{v}_i) = d_i j \vec{w}_j \tag{3}$$

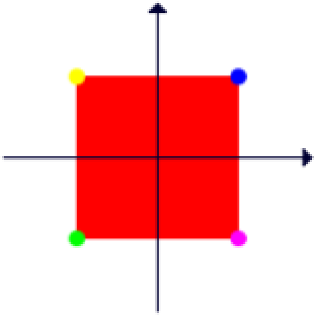
이에 착안해서, d_{ij}를 이용해서 행렬 D를 만들면, n by m 행렬이 된다. 이 행렬을 Transformation Matrix라고 한다. 이 행렬을 이용하면, $\vec{v} \in V$에 대해서 $f(\vec{v}) = D\vec{v}$ 임을 알 수 있다.

선형변환의 예시 : 확대/축소



선형변환의 예시 : 반전



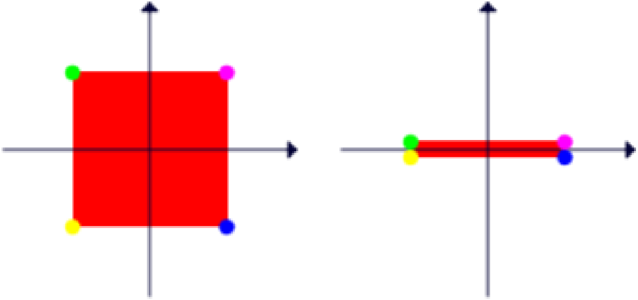


선형변환의 예시 : 확대/축소

$$T_\theta = \begin{bmatrix} p & 0 \\ 0 & q \end{bmatrix}$$

- $p, q > 1$: 확대
- $0 < p, q < 1$: 축소
- $pq < 0$: 반전 (선대칭)
- $p < 0, q < 0$: 반전 (점대칭)

선형변환의 예시 : 사영

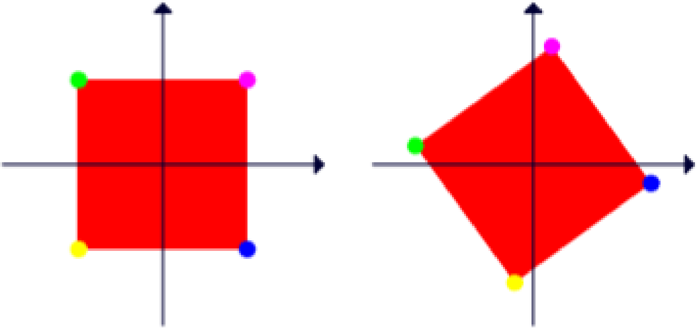


선형변환의 예시 : 사영

$$T_\theta = \begin{bmatrix} x & 0 \\ 0 & y \end{bmatrix}$$

- $x = 1, y = 0$: x축 사영
- $x = 0, y = 1$: y축 사영

선형변환의 예시 : 회전변환



선형변환의 예시 : 회전변환

$$T_\theta = \begin{bmatrix} cos\theta & -sin\theta \\ sin\theta & cos\theta \end{bmatrix}$$

- $\theta > 0$: 반시계방향으로 θ만큼 회전
- $\theta < 0$: 시계방향으로 θ만큼 회전

선형변환의 예시 : Orthogonal Transformation

선형변환 $T : V \to W$ 에 대해서, V의 임의의 원소 $\vec{v}, \vec{w}$에 대해서 다음을 만족하면 Orthogonal Transformation이라고 하며, 이 때 T에 대응되는 행렬을 orthogonal matrix라고 한다.
$\vec{v} \bullet \vec{u} = T\vec{v} \bullet T\vec{u}$
즉, Orthogonal Matrix는 벡터의 크기를 보존시키며
$(|\vec{v}|^2 = \vec{v} \bullet \vec{v} = T\vec{v} \bullet T\vec{v} = |T\vec{v}|^2)$, 벡터 간의 각도를 보존한다. Orthogonal Transformation은 회전변환이나, 회전변환 후 반전을 하는 선형변환이다.

Orthogonal Matrix

n by n Orthogonal Matrix A는 다음의 특징을 가진다.

- $A^T A = AA^T = I$ [1]
- $det(A) = 1$ or $det(A) = -1$ 이다. 역은 성립하지 않는다. 1인 경우, A는 회전변환이며 -1인 경우 A는 회전변환과 반전이다.
- A의 행벡터/열벡터는 $\mathbb{R}^n$의 기저이며, 각 벡터의 크기는 모두 1이다.

[1] 이 성질이 orthogonal matrix의 정의로 보기도 한다.

행렬의 연산을 이용한 선형변환

선형변환이라는 함수가 행렬으로 표현가능하므로, 행렬의 연산을 이용해서 선형변환을 원하는 대로 만들 수 있다. 여기서는 크게 두 가지를 살펴보고자 한다.

- 합성함수와 행렬의 곱
- 역함수와 역행렬

이 두 가지 예시를 회전변환을 이용해서 살펴보고자 한다.

행렬 곱과 합성함수

두 회전변환

$$T_\alpha = \begin{bmatrix} cos\alpha & -sin\alpha \\ sin\alpha & cos\alpha \end{bmatrix}$$

$$T_\beta = \begin{bmatrix} cos\beta & -sin\beta \\ sin\beta & cos\beta \end{bmatrix}$$

가 있을 때, 두 변환행렬의 곱은

$$T_\alpha T_\beta = \begin{bmatrix} cos\alpha & -sin\alpha \\ sin\alpha & cos\alpha \end{bmatrix} \times \begin{bmatrix} cos\beta & -sin\beta \\ sin\beta & cos\beta \end{bmatrix} =$$

$$\begin{bmatrix} cos\alpha cos\beta - sin\alpha sin\beta & -cos\alpha sin\beta - sin\alpha cos\beta \\ cos\alpha sin\beta + sin\alpha cos\beta & cos\alpha cos\beta - sin\alpha sin\beta \end{bmatrix}$$

인데, 이는 삼각함수의 합차공식을 이용하면 $T_{\alpha+\beta}$임을 알 수 있다.

역행렬과 역함수

T_α의 역행렬은 다음과 같이 구할 수 있다.

$T_\alpha T^{-1} = \begin{bmatrix} cos\alpha & -sin\alpha \\ sin\alpha & cos\alpha \end{bmatrix} \times \begin{bmatrix} cos\alpha & sin\alpha \\ -sin\alpha & cos\alpha \end{bmatrix} = I_2$ 인데, 이에서

$T^{-1} = T_{-\alpha}$임을 알 수 있다.

Definition of Eigen*

행렬 A에 대해서, 다음을 만족하는 λ와 $\vec{x}$들을 각각 고유값(eigenvalue)과 고유벡터(eigenvector)라고 한다.

$A\vec{x} = \lambda\vec{x}$ 위 식을 다시 쓰면 $(A - \lambda I)\vec{x} = 0$ 이며, 여기서 $det(A - \lambda I) = 0$

를 특성방정식이라 한다. 이 때 특성방정식의 해는 고유값이 되며, 이에 따라 고유벡터를 계산할 수 있다. 고유벡터들로 span되는 공간을

eigenspace라 한다.

위에서 다룬 회전변환 행렬 T를 생각해 보자.

$$T_\theta = \begin{bmatrix} cos\theta & -sin\theta \\ sin\theta & cos\theta \end{bmatrix}$$

이 때, 회전의 축을 어떻게 구할 수 있을까? 회전의 축은 회전 전후에도 변화가 없을 것이다. 따라서, 회전의 축을 벡터 $\vec{a}$라 하면, $T\vec{a} = \vec{a}$ 임이 성립해야 한다. 이 경우, 이를 만족하는 벡터 a는 0벡터 뿐이다. 그렇다면 이는 무슨 의미를 가질까? 이는 회전축이 z축이므로, 이를 나타내는 것으로 생각할 수 있다.

이는 다음의 3차원 회전변환 행렬을 생각해보면 조금 더 명확해진다.

$$T_\theta = \begin{bmatrix} cos\theta & -sin\theta & 0 \\ sin\theta & cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

이 경우 위와 같은 방정식을 풀면 z는 어떠한 수여도 가능하고, x와 y는 0임을 알 수 있다. 즉, z축이 회전의 축임을 알 수 있다.
이는 일반적인 선형변환에서도 같다. 어떤 선형변환 R에 대해서, $R\vec{a} = \vec{a}$가 성립한다면, 그 벡터는 선형변환에 대해서 불변이다. 즉, 어떠한 행렬의 eigenvector는 그 행렬에 대응하는 선형변환의 축이라고 볼 수 있다.

Properties of Eigen*

일반적으로, 다음 성질들이 성립한다.

- A^T의 고유값과 고유벡터는 A와 같다.
- $A^T A$의 고유벡터는 A와 같고, 고유값은 제곱값이다.
- det(A)는 모든 고유값의 곱이다.

Proof of 3) I

어떤 행렬 A의 고유벡터를 $\vec{v_i}$, 고유값을 λ_i라 하자. 그러면 다음이 성립한다.

$$
\begin{aligned}
A\begin{bmatrix}\vec{v_1} & \vec{v_2} & \dots & \vec{v_n}\end{bmatrix} &= A\begin{bmatrix}A\vec{v_1} & A\vec{v_2} & \dots & A\vec{v_n}\end{bmatrix} & (4)\\
&= A\begin{bmatrix}\lambda_1\vec{v_1} & \lambda_2\vec{v_2} & \dots & \lambda_n\vec{v_n}\end{bmatrix} & (5)\\
&= \begin{bmatrix}\lambda_1\vec{v_1} & \lambda_2\vec{v_2} & \dots & \lambda_n\vec{v_n}\end{bmatrix}\begin{bmatrix}\lambda_1 & 0 & \dots & 0\\ 0 & \lambda_2 & \dots & 0\\ 0 & 0 & \dots & \lambda_n\end{bmatrix} & (6)
\end{aligned}
$$

이다. 이 때, $\det(AB) = \det(A)\det(B)$임을 이용하면

Proof of 3) II

$$det(A)det\left(\begin{bmatrix}\vec{v}_1 & \vec{v}_2 & \dots & \vec{v}_n\end{bmatrix}\right) \tag{7}$$

$$= det\left(\begin{bmatrix}\lambda_1\vec{v}_1 & \lambda_2\vec{v}_2 & \dots & \lambda_n\vec{v}_n\end{bmatrix}\right) det\left(\begin{bmatrix}\lambda_1 & 0 & \dots & 0\\ 0 & \lambda_2 & \dots & 0\\ 0 & 0 & \dots & \lambda_n\end{bmatrix}\right) \tag{8}$$

$$\tag{9}$$

이므로, det(A)는 모든 고유값의 곱이 된다.

Eigendecomposition

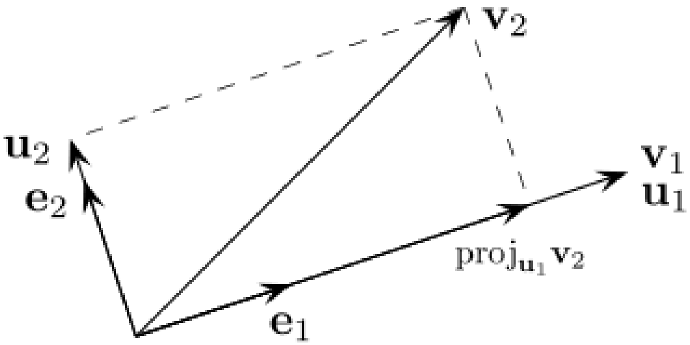
Eigendecomposition은 어떤 행렬의 eigenvector들이 서로 선형독립일 때 가능하다.

$$A\begin{bmatrix}\vec{v}_1 & \vec{v}_2 & \dots & \vec{v}_n\end{bmatrix} = \begin{bmatrix}\lambda_1\vec{v}_1 & \lambda_2\vec{v}_2 & \dots & \lambda_n\vec{v}_n\end{bmatrix}\begin{bmatrix}\lambda_1 & 0 & \dots & 0\\ 0 & \lambda_2 & \dots & 0\\ 0 & 0 & \dots & \lambda_n\end{bmatrix} \tag{10}$$

에서, 좌변의 $\begin{bmatrix}\vec{v}_1 & \vec{v}_2 & \dots & \vec{v}_n\end{bmatrix}$ 가 만약 역행렬을 가진다면, 다음과 같이 A를 분해할 수 있다.

$A = QLQ^{-1}$

여기서 Q는 $\begin{bmatrix}\vec{v}_1 & \vec{v}_2 & \dots & \vec{v}_n\end{bmatrix}$ 이고, L은 $\begin{bmatrix}\lambda_1 & 0 & \dots & 0\\ 0 & \lambda_2 & \dots & 0\\ 0 & 0 & \dots & \lambda_n\end{bmatrix}$ 이다.



선형독립인 두 벡터 $\vec{v}_1, \vec{v}_2$에 대해서 $proj_{\vec{v}_1}\vec{v}_2$는 다음과 같이 정의된다.

$\frac{\vec{v}_1 \bullet \vec{v}_2}{\vec{v}_1 \bullet \vec{v}_1}\vec{v}_1$

그림에서 볼 수 있듯이, 이는 벡터 $\vec{v}_2$의 $\vec{v}_1$ 방향으로의 성분을 나타낸다.

Projection II

또한, $(\vec{v}_2 - proj_{\vec{v}_1}\vec{v}_2) \bullet \vec{v}_1 = 0$ 이다. 이에서 보면 알 수 있듯이, 사영을

통해서 두 선형독립인 벡터들 $V = \{\vec{v}_1, \vec{v}_2\}$를 서로 수직인 벡터 $W = \{\vec{v}_1, proj_{\vec{v}_1}\vec{v}_2\}$으로 바꿀 수 있다. 또한 W의 원소들을 각각 그 원소의 크기로 나눔으로써 크기가 1이고 서로 수직한 벡터로 만들 수 있다.

위 과정을 선형독립인 벡터 n개에 대해서 다음과 같이 시행하는 과정을 Gram-Schmidt Process라고 한다. 즉, 선형독립인 벡터 $\{\vec{v_i}\}$에 대해서
$\vec{u_1} = \vec{v_1}, \vec{e_1} = \frac{\vec{u_1}}{|\vec{u_1}|}$
$\vec{u_2} = \vec{v_2} - proj_{\vec{v_1}}\vec{v_2}, \vec{e_2} = \frac{\vec{u_2}}{|\vec{u_2}|}$
$\vec{u_3} = \vec{v_3} - proj_{\vec{v_1}}\vec{v_3} - proj_{\vec{v_2}}\vec{v_3}, \vec{e_3} = \frac{\vec{u_3}}{|\vec{u_3}|}$
와 같이 $\{\vec{e_i}\}$ 를 만들 수 있다. 이 때, 기존의 벡터 $\vec{v_i}$를 새로 얻은 벡터들 $\vec{e_i}$ 로 다음과 같이 나타낼 수 있다.

$$\vec{v_1} = \vec{v_1} \bullet \vec{e_1}\vec{e_1}$$
$$\vec{v_2} = \vec{v_1} \bullet \vec{e_1}\vec{e_1} + \vec{v_1} \bullet \vec{e_2}\vec{e_2}$$
$$\vec{v_3} = \vec{v_3} \bullet \vec{e_1}\vec{e_1} + \vec{v_3} \bullet \vec{e_2}\vec{e_2} + \vec{v_3} \bullet \vec{e_3}\vec{e_3}$$

이를 행렬을 이용하여 다시 써 보면, 다음과 같이 나타낼 수 있다.

Gram-Schmidt Process II

$V = QR$
여기서,

$V = [\vec{v_1}\,\vec{v_2}\ldots\vec{v_n}]$
$Q = [\vec{e_1}\,\vec{e_2}\ldots\vec{e_n}]$

$$R = \begin{bmatrix} \vec{e_1}\bullet\vec{v_1} & \vec{e_1}\bullet\vec{v_2} & \vec{e_1}\bullet\vec{v_3} & \ldots & \vec{e_1}\bullet\vec{v_n} \\ 0 & \vec{e_2}\bullet\vec{v_2} & \vec{e_2}\bullet\vec{v_3} & \ldots & \vec{e_2}\bullet\vec{v_n} \\ 0 & 0 & \vec{e_3}\bullet\vec{v_3} & \ldots & \vec{e_3}\bullet\vec{v_n} \\ \ldots & \ldots & \ldots & \ldots & \ldots \end{bmatrix}$$

QR decomposition

위에서와 같이 Gram-Schmidt 과정을 거쳐, Orthonormal[2]한 벡터집합을 만들 수 있으며, 이 과정에서 위 식 $V = QR$과 같이 어떤 행렬을 두 행렬의 곱으로 분해할 수 있다. 이러한 분해를 QR decomposition이라고 한다. 이 때 이러한 분해가 가능하기 위해서는

- V의 각 column 벡터들이 서로 선형 독립이여야 한다.

[2] 서로 orthogonal하며, 모두의 크기가 1인 벡터 집합

일차방정식 Solver 만들기

일차방정식들 $a_{ij}x_j = b_i, i, j = 1, 2, ..., n$을 다음과 같이 쓸 수 있다.
$A\vec{x} = \vec{b}$
여기서 $A_{ij} = a_{ij}, \vec{x}[i] = x_i, \vec{b}[i] = b_i$이다. 이때, 일차방정식의 해 $\vec{x}$는 다음과 같은 과정을 통해서 구할 수 있다.

- A의 역행렬을 구한다.
    - 만약 실패하면 해가 없거나 무한히 많은 것이다.
    - 성공하면, $A^{-1}\vec{b}$ 를 계산한다.

일차방정식 Solver 만들기 : QR decomposition을 이용

$A\vec{x} = \vec{b}$
여기서 $A_{ij} = a_{ij}, \vec{x}[i] = x_i, \vec{b}[i] = b_i$이다. 이때, 일차방정식의 해 $\vec{x}$는 다음과 같은 과정을 통해서 구할 수 있다.

- A = QR로 분해한다.
  - 만약 실패하면 해가 없거나 무한히 많은 것이다.
  - 성공하면, $A\vec{x} = \vec{b}$ 양변에 Q^T를 곱한다. Q는 orthogonal matrix이므로 $QQ^T = I$이다. 따라서 $R\vec{x} = Q^T\vec{b}$ 이다. R은 삼각행렬이므로 어렵지 않게 $\vec{x}$를 구할 수 있다.

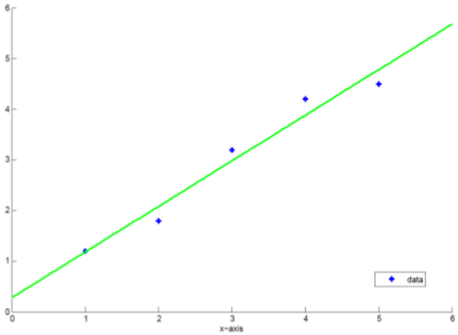
위에서는 n by n 행렬 A와 n벡터 $\vec{x}, \vec{b}$에 대해서 $A\vec{x} = \vec{b}$인 경우에, $\vec{x}$를 찾는 방법을 알아보았다. 이번에는 m by n 행렬 A와 m벡터 $\vec{b}$에 대해서 $A\vec{x} \approx \vec{b}$를 만족하는 n벡터 $\vec{x}$를 찾아보자. [3]
여기서, $\approx$의 기준을 다음과 같이 잡자.

$$min_{\vec{x}}|A\vec{x} - \vec{b}|^2$$

즉, $|A\vec{x} - \vec{b}|^2$ 값을 최소로 하는 $\vec{x}$를 찾아보자. 이러한 문제를 Linear Least Squares라 한다.

[3] 만약 m=n인 경우라면, 위의 선형방정식 풀이 문제가 되므로 이 문제는 선형방정식 풀이의 일반화된 문제라고 볼 수 있다.

Motivation : Linear Regression



데이터 $(x_i, y_i), i = 1, 2, ..., n$이 주어졌을 때, 이 데이터를 가장 잘 설명하는 $y(x) = ax + b$를 찾아보자.

이 때, 문제를 다음과 같이 볼 수 있다.

$\sum_{i=1}^{n}(ax_i + b - y_i)^2$이 최소가 되는 a,b는 무엇일까?

이 문제를 행렬로 풀어 쓰면, 다음과 같이 생각할 수 있다.

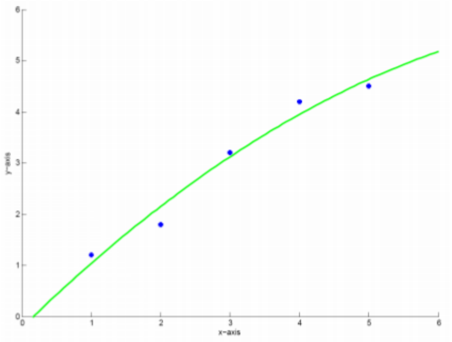
$$A = \begin{bmatrix} x_1 & 1 \\ x_2 & 1 \\ \dots & \dots \\ x_n & 1 \end{bmatrix} \in \mathbb{R}^{m\times 2}, \vec{y} = \begin{bmatrix} y_1 \\ y_2 \\ \dots \\ y_n \end{bmatrix} \in \mathbb{R}^{m}$$

라고 하면,

$\left| A \begin{bmatrix} a \\ b \end{bmatrix} - \vec{y} \right|^2$ 를 최소화하는 문제가 된다. 즉, 저 직선을 예측하는 문제는 곧 LLS 문제를 푸는 것이 된다.

이제, 위 문제를 조금 더 확장해 보자.

Motivation : Polynomial Regression



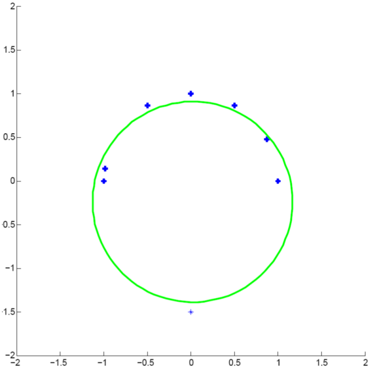
데이터 $(x_i, y_i), i = 1, 2, ..., n$이 주어졌을 때, 이 데이터를 가장 잘 설명하는 p차함수 $y(x) = a_i x^i$를 찾아보자.

위와 비슷하게, 다음 LLS 문제로 볼 수 있다.

$$\left\| \begin{bmatrix} x_1^p & x_1^{p-1} & \dots & x_1 & 1 \\ x_2^p & x_2^{p-1} & \dots & x_2 & 1 \\ \dots & \dots & \dots & \dots & \dots \\ x_n^p & x_n^{p-1} & \dots & x_n & 1 \end{bmatrix} \begin{bmatrix} a_p \\ a_{p-1} \\ \dots \\ a_1 \\ a_0 \end{bmatrix} - \begin{bmatrix} y_1 \\ y_2 \\ \dots \\ y_n \end{bmatrix} \right\|^2$$

을 최소로 만드는 a_i는?

Motivation : Curve Regression



데이터 $(x_i, y_i), i = 1, 2, ..., n$이 주어졌을 때, 가장 적절한 $(x - c_1)^2 + (y - c_2)^2 = r^2$을 찾아라.

이 문제 역시 LLS 문제로 볼 수 있다. 위에서 원의 방정식을 전개하여 정리하면 $2xc_1 + 2yc_2 + (r^2 - c_1^2 - c_2^2) - (x^2 + y^2) = 0$ 이므로, $c_3 = r^2 - c_1^2 - c_2^2$이라고 하면 다음과 같은 LLS 문제가 된다.

$$\left\| \begin{bmatrix} 2x_1 & 2y_1 & 1 \\ 2x_2 & 2y_2 & 1 \\ \cdots & \cdots & \cdots \\ 2x_n & 2y_n & 1 \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \\ c_3 \end{bmatrix} - \begin{bmatrix} x_1^2 + y_1^2 \\ x_2^2 + y_2^2 \\ \cdots \\ x_n^2 + y_n^2 \end{bmatrix} \right\|^2$$

을 최소로 만드는 c_i는?

어떤 $\vec{x*}$이 LLS 문제 $min_{\vec{x}}|A\vec{x} - \vec{b}|^2$ 의 해라고 하자. 그렇다면 다음이 성립한다.

$$|A\vec{x*} - \vec{b}|^2 = min_{\vec{x}}|A\vec{x} - \vec{b}|^2 \tag{11}$$

그렇다면, 임의의 $\vec{y} \in \mathbb{R}^n$에 대해서,

$$|A\vec{x*} - \vec{b}|^2 \leq |A(\vec{x*} + \vec{y}) - \vec{b}|^2 \tag{12}$$

인데, 이 식의 좌변을 $|\vec{x}|^2 = \vec{x} \bullet \vec{x}$임을 이용하여 전개하면 다음과 같다.

$$
\begin{aligned}
& |A(\vec{x*}+\vec{y})-\vec{b}|^2 \\
= \;& (A(\vec{x*}+\vec{y})-\vec{b})^T(A(\vec{x*}+\vec{y})-\vec{b}) \\
= \;& ((\vec{x*}+\vec{y})^T A^T-\vec{b}^T)(A(\vec{x*}+\vec{y})-\vec{b}) \\
= \;& (\vec{x*}+\vec{y})^T A^T A(\vec{x*}+\vec{y})-\vec{b}^T A(\vec{x*}+\vec{y})-(\vec{x*}+\vec{y})^T A^T\vec{b}+\vec{b}^T\vec{b} \\
= \;& \vec{x*}^T A^T A\vec{x*}-2\vec{x*}^T A^T\vec{b}+\vec{b}^T\vec{b}+2\vec{y}^T A^T A\vec{x*}-2\vec{y}^T A^T\vec{b}+\vec{y}^T A^T\vec{y} \\
= \;& |A\vec{x*}-\vec{b}|^2+2\vec{y}^T(A^T A\vec{x*}-A^T\vec{b})+|A\vec{y}|^2
\end{aligned}
$$

이다. 따라서 1) $0 \leq 2\vec{y}^T(A^T A\vec{x*}-A^T\vec{b})+|A\vec{y}|^2$ 이여야 한다.
이 때, $|A\vec{y}|^2$은 0 이상이므로 첫 번째 항 $2\vec{y}^T(A^T A\vec{x*}-A^T\vec{b})$을 보자. 만약 이 항이 충분히 큰 음수라면, 1)은 성립하지 않는다. 여기서, $\vec{y}$가 사실 다음의 조건 2)를 만족하는 벡터였다고 생각하자.

$$\vec{y} = -\alpha(A^T A\vec{x*}-A^T\vec{b}) \tag{13}$$

만약 그렇더라도, 임의의 벡터 $\vec{y}$에 대해서 1)이 다 성립해야 하므로 2)를 1)에 대입하여도 성립하여야 한다. 대입하여 계산하면 다음과 같은 결과가 나온다.

$$2\vec{y}^T(A^TA\vec{x*} - A^T\vec{b}) + |A\vec{y}|^2 \quad (14)$$

$$= -2\alpha|A^TA\vec{x*} - A^T\vec{b}|^2 + \alpha^2|A(A^TA\vec{x*} - A^T\vec{b}|^2 \quad (15)$$

이 된다. 이 식은 α에 대한 이차함수로 볼 수 있다. 이 때, 이 식이 항상 0 이상이 될 조건은 $a > 0$일 때 $ax^2 - bx$가 0 이상일 조건과 같다. 즉, 판별식 $D = b^2 - 4ac = b^2$ 이 0 이하여야 한다. 따라서 b는 0이여야만 하는데, 원식에서 b는 $|A^TA\vec{x*} - A^T\vec{b}|$ 이므로, LLS를 만족시키는 해 $\vec{x*}$는

$$A^TA\vec{x*} - A^T\vec{b} = \vec{0} \quad (16)$$

이여야 한다.

Solving LLS : Theory IV

역으로, 어떤 벡터 $\vec{x*}$이 $A^TA\vec{x*} - A^T\vec{b} = \vec{0}$을 만족한다면, 임의의 벡터 $\vec{x}$에 대해서 $\vec{y} = \vec{x} - \vec{x*}$이라 하면 다음이 성립한다.

$$
\begin{aligned}
|A\vec{x} - \vec{b}|^2 &= |A\vec{x*} + A\vec{y} - \vec{b}|^2 && (17)\\
&= |A\vec{x*} - b|^2 + 2\vec{y}^T(A^TA\vec{x*} - A^T\vec{b}) + |A\vec{y}|^2 && (18)\\
&= |A\vec{x*} - b|^2 + |A\vec{y}|^2 && (19)\\
&\geq |A\vec{x*} - b|^2 && (20)
\end{aligned}
$$

따라서, $\vec{x*}$는 LLS의 해이다.

Solution of LLS

위 슬라이드에서 볼 수 있듯이, LLS의 해는 $A^T A\vec{x} - A^T\vec{b} = 0$을 만족하는 $\vec{x}$이다. 이제 이를 구하는 방법을 생각해 보자. 크게 두 가지 방법을 생각해볼 수 있다.

- 선형방정식 풀이를 이용[4]
- QR Decomposition의 적용

선형방정식의 풀이는 이미 해보았으므로, QR Decomposition을 적용해 보겠다.

[4]위 식에서 $A^T A$를 행렬 A의 Psuedoinverse라 한다.

Solving LLS with QR decomposition I

먼저, $A = Q \begin{bmatrix} R \\ 0 \end{bmatrix}$으로 분해하자. 이 때 Q는 orthogonal matrix이므로 다음이 성립한다.

- $Q^T A = \begin{bmatrix} R \\ 0 \end{bmatrix}$
- $|(A\vec{(x)} - \vec{b})|^2 = |Q^T(A\vec{(x)} - \vec{b})|^2$

가 성립한다.
여기서 두 번째 식에 초점을 맞추어 보면, 다음과 같은 계산이 가능하다.

$$
\begin{aligned}
|(A\vec{(x)} - \vec{b})|^2 &= |Q^T(A\vec{(x)} - \vec{b})|^2 && (21) \\
&= |Q^T A\vec{(x)} - \vec{b})|^2 && (22) \\
&= \left| \begin{bmatrix} R \\ 0 \end{bmatrix} \vec{(x)} - Q^T \vec{b}) \right|^2 && (23)
\end{aligned}
$$

Solving LLS with QR decomposition II

여기서 $Q^T\vec{b} = \begin{bmatrix} c \\ d \end{bmatrix}$ 로 분할하면 위 식은 다음과 같이 바뀐다.

$$
\begin{aligned}
|A\vec{x} - \vec{b}|^2 &= \left| \begin{bmatrix} R \\ 0 \end{bmatrix} \vec{x} - \begin{bmatrix} c \\ d \end{bmatrix}) \right|^2 && (24) \\
&= \left| \begin{bmatrix} R\vec{x} - c \\ -d \end{bmatrix} \right|^2 && (25) \\
&= |R\vec{x} - c|^2 + |d|^2 && (26)
\end{aligned}
$$

이므로, $|R\vec{x} - c|^2 = 0$일 때 $|A\vec{x} - \vec{b}|^2$이 최소가 된다. 따라서 $\vec{x} = R^{-1}c$가 된다. 따라서

$$\vec{x} = R^{-1}c \tag{27}$$

이다.

Back to the Motivation : Linear Regression

Coding!